e-merge: A Heuristic for Improving Minimum Power Broadcast Trees in Wireless Networks

Arindam K. Das, Robert J. Marks, Mohamed El-Sharkawi, Payman Arabshahi, Andrew Gray

Abstract— Wireless multicast/broadcast sessions, unlike wired networks, inherently reaches several nodes with a single transmission. For omnidirectional wireless broadcast to a node, all nodes closer will also be reached. Heuristic algorithms for constructing the minimum power tree in wireless networks have been proposed by Wieselthier *et al* [1] and Stojmenovic *et al* [4]. In this paper, we present the e-merge procedure, a heuristic for improving minimum power broadcast (MPB) trees in wireless networks. Simulation results show that better solutions are usually obtained, with considerably lower tree power, if the proposed procedure is applied to the trees generated using the Broadcast Incremental Power (BIP) algorithm discussed in [1].

I. INTRODUCTION

Unlike wired networks, where a transmission $i \rightarrow j$ reaches only node j, in wireless networks, several nodes may be reached by a single transmission. Assuming that all nodes have omnidirectional antennas, nodes which are closer to i than j will also receive the transmission directed to j. This is the "wireless advantage property" [1]. Because of this property, if the source wants to broadcast to all other nodes in the network, it can do so simply by transmitting to the node farthest from it. Similarly, the minimum spanning tree (MST) of the network, where each node is actually connected to some other node, is also a valid wireless broadcast tree. Clearly, while the wireless advantage property is utilized to the fullest in the single transmission solution, it is not utilized at all in the MST solution. Given the MST, or any valid broadcast tree, the e-merge ($e \geq 2$) procedure described in this chapter attempts to improve the solution by utilizing the wireless advantage property and replacing a block of 'e' successive transmissions by a cheaper single transmission, while ensuring that the connectivity of the tree is not affected. Connectivity in a wireless broadcast tree is ensured if the kth ($\forall k \geq 2$; for $k = 1$, the transmitting node is the source) transmitting node in the tree has been reached by any of the prior transmissions.

The power matrix and reward matrix of an N-node network are defined as follows:

- For any N-node network, the power matrix, $\mathbf{P}$, is an $N \times N$ symmetric matrix. The $(i,j)th$ element of the power matrix represents the power required for node i to transmit to node j and is given by:

$$\mathbf{P}_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2\right]^{\alpha/2} = d_{ij}^{\alpha}$$

where $\{(x_i, y_i) : 1 \leq i \leq N\}$ are the coordinates of the nodes in the network, α ($\alpha \geq 2$) is the channel loss exponent and d_{ij} is the Euclidean distance between nodes i and j.

- The reward matrix, $\mathbf{R}$, of a network is a representation of the nodes covered (or not) by all possible transmissions in the network. In MATLAB© notation, $\mathbf{R}$ is a cell array, each cell being an N-element binary vector. Using the notation $\mathbf{R}_{ij}(p)$ to index the pth element of the (i,j) cell in $\mathbf{R}$, we compute the reward matrix as follows:

$$\mathbf{R}_{ij}(p) = \begin{cases} 1, & \text{if } \mathbf{P}_{ij} \leq \mathbf{P}_{ij} \\ 0, & \text{otherwise} \end{cases}$$

For example, in the network shown in Figure 1, the transmission $2 \rightarrow 1$ will cover nodes 1, 5 and 6. This information is encoded in the (2,1) cell of the reward matrix as: $\mathbf{R}_{21} = [1\ 0\ 0\ 0\ 1\ 1]$.

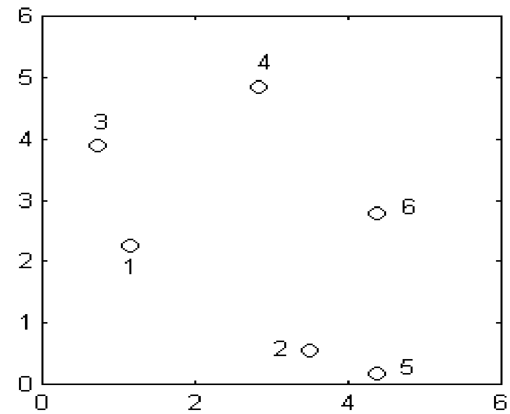

Fig. 1. Example 6-node network

The power matrix and the reward matrix of the network shown in Figure 1, assuming $\alpha = 2$, are:

$$\mathbf{P} = \begin{bmatrix} 0 & 8.51 & 2.79 & 9.51 & 14.92 & 10.73 \\ 8.51 & 0 & 18.80 & 19.02 & 0.93 & 5.75 \\ 2.79 & 18.80 & 0 & 5.29 & 27.18 & 14.51 \\ 9.51 & 19.02 & 5.29 & 0 & 24.48 & 6.74 \\ 14.92 & 0.93 & 27.18 & 24.48 & 0 & 6.85 \\ 10.73 & 5.75 & 14.51 & 6.74 & 6.85 & 0 \end{bmatrix} \quad (1)$$

A.K. Das, R.J. Marks and M. El-Sharkawi are with Department of Electrical Engineering, University of Washington, Box 352500, Seattle, WA 98195. e-mails: {arindam, marks, melshark}@ee.washington.edu.

P. Arabshahi and A. Gray are with Jet Propulsion Laboratory, 4800 Oak Grove Drive, MS 238-343, Pasadena, CA 91109. e-mails: {payman, gray}@arcadia.jpl.nasa.gov

$$\mathbf{R} = \begin{bmatrix} [0\;0\;0\;0\;0\;0] & [0\;1\;1\;0\;0\;0] & [0\;0\;1\;0\;0\;0] & [0\;1\;1\;1\;0\;0] & [0\;1\;1\;1\;1\;1] & [0\;1\;1\;1\;0\;1] \\ [1\;0\;0\;0\;1\;1] & [0\;0\;0\;0\;0\;0] & [1\;0\;1\;0\;1\;1] & [1\;0\;1\;1\;1\;1] & [0\;0\;0\;0\;1\;0] & [0\;0\;0\;0\;1\;1] \\ [1\;0\;0\;0\;0\;0] & [1\;1\;0\;1\;0\;1] & [0\;0\;0\;0\;0\;0] & [1\;0\;0\;1\;0\;0] & [1\;1\;0\;1\;1\;1] & [1\;0\;0\;1\;0\;1] \\ [1\;0\;1\;0\;0\;1] & [1\;1\;1\;0\;0\;1] & [0\;0\;1\;0\;0\;0] & [0\;0\;0\;0\;0\;0] & [1\;1\;1\;0\;1\;1] & [0\;0\;1\;0\;0\;1] \\ [1\;1\;0\;0\;0\;1] & [0\;1\;0\;0\;0\;0] & [1\;1\;1\;1\;0\;1] & [1\;1\;0\;1\;0\;1] & [0\;0\;0\;0\;0\;0] & [0\;1\;0\;0\;0\;1] \\ [1\;1\;0\;1\;1\;0] & [0\;1\;0\;0\;0\;0] & [1\;1\;1\;1\;1\;0] & [0\;1\;0\;1\;0\;0] & [0\;1\;0\;1\;1\;0] & [0\;0\;0\;0\;0\;0] \end{bmatrix} \quad (2)$$

II. COST REDUCTION MECHANISMS

The e-merge procedure employs three cost-reduction[1] mechanisms. These are:

1) Replacing multiple transmissions from a node by the highest-powered transmission (procedure MTR).
2) Elimination of redundant transmissions, or edge trimming (procedure ET).
3) Merging a block of e ($e \geq 2$) successive transmissions by a cheaper single transmission (procedure EM).

Before describing the above mechanisms, we will define the following sets:

k = transmission step number

$\mathbf{NA}^{(k)}$ = set of all nodes reached in transmission step k ($\mathbf{NA}^{(0)}$ = [source])

$\mathbf{NN}^{(k)}$ = set of new nodes reached in transmission step k ($\mathbf{NN}^{(0)}$ = [source])

$\mathbf{NA}^{(0:k)}$ = set of all nodes reached till transmission step k

Note that:

$$\mathbf{NA}^{(0:k)} = \bigcup_{m=0}^{k} \mathbf{NA}^{(m)} = \bigcup_{m=0}^{k} \mathbf{NN}^{(m)} \quad (3)$$

a. Procedure MTR: Multiple transmissions from a node are unnecessary in wireless networks because the highest-powered transmission also reaches other nodes which are covered by lower-powered transmissions. Given a sequence of transmissions constituting a broadcast tree, the first step in cost reduction is therefore to replace the first transmission from a node by the highest powered transmission appearing later in the sequence. All other transmissions from the node can then be deleted from the sequence. For example, referring to Figure 1 and assuming that node 6 is the source, let the starting solution and its cost be:

$$\text{Starting solution} = \{6 \to 2, 2 \to 5, 6 \to 4, 4 \to 3, 3 \to 1\} \quad (4)$$

$$\text{Cost} = 21.50 \quad (5)$$

Since there are two transmissions from node 6, we can replace the lower-powered transmission $6 \to 2$ by the higher-powered transmission $6 \to 4$. The resulting tree and its cost are then:

$$\text{Improved solution} = \{6 \to 4, 2 \to 5, 4 \to 3, 3 \to 1\} \quad (6)$$

$$\text{Cost} = 15.75 \quad (7)$$

[1]Throughout this paper, cost means overall tree power.

b. Procedure ET: The kth transmission in a broadcast tree is redundant if no new node is reached by it; or equivalently, if the nodes reached by it have already been reached by the prior transmissions. The conditions for kth step transmission redundancy are therefore:

$$\mathbf{NN}^{(k)} = \emptyset \iff \mathbf{NA}^{(k)} \subseteq \mathbf{NA}^{(0:k-1)} \quad (8)$$

For the network in Figure 1, suppose the starting solution is:

$$\text{Starting solution} = \{6 \to 3,\ 3 \to 1\} \quad (9)$$

$$\text{Cost} = 17.30 \quad (10)$$

The nodes reached by the transmissions in (9) are shown in Table I. The second column in the table lists the $\{$*transmitting node*, *destination node*$\}$ pairs in the solution.

TABLE I

Nodes reached by the transmissions in (9).

k	$t \to d$	$\mathbf{NA}^{(k)}$	$\mathbf{NN}^{(k)}$	$\mathbf{NA}^{(0:k)}$
0	-	6	6	6
1	$6 \to 3$	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5,6
2	$3 \to 1$	1	$\emptyset$	1,2,3,4,5,6

Note that the elements in the set $\mathbf{NA}^{(1)}$ are the node indices corresponding to '1' in the cell $\mathbf{R}_{63}$ (2). Similarly, the element in the set $\mathbf{NA}^{(2)}$ is the node index corresponding to '1' in the cell $\mathbf{R}_{31}$. Since node 1 has already been reached by the transmission at step 1, the set $\mathbf{NN}^{(2)}$ is empty and hence, the transmission $3 \to 1$ is redundant. An improved solution is therefore:

$$\text{Improved solution} = \{6 \to 3\} \quad (11)$$

$$\text{Cost} = 14.51 \quad (12)$$

c. Procedure EM: Given a transmission sequence t_1, t_2, $\cdots$ t_K, a block of 2 ($e = 2$) successive transmissions, say t_k and t_{k+1}, is replaced by a single transmission t'_k if:

1) the cost of t'_k is smaller than the sum of the costs of t_k and t_{k+1}.
2) the set of new nodes reached by t_k and t_{k+1} is a subset of the set of all nodes reached by t'_k.
3) the transmitting node in t'_k is the source or any other node which has already been reached by the prior transmissions t_1 to t_{k-1}. That is, the transmitting node in t'_k must be an element of the set $\mathbf{NA}^{(0:k-1)}$.

For the network in Figure 1, suppose the starting solution is:

$$\text{Starting solution} = \{6 \to 4, 2 \to 5, 4 \to 3, 3 \to 1\} \quad (13)$$
$$\text{Cost} = 15.75 \quad (14)$$

The nodes reached by the transmissions in the starting solution are shown in Table II.

TABLE II

Nodes reached by the transmissions in (13).

k	$t \to d$	$\mathbf{NA}^{(k)}$	$\mathbf{NN}^{(k)}$	$\mathbf{NA}^{(0:k)}$
0	-	6	6	6
1	$6 \to 4$	2,4	2,4	2,4,6
2	$2 \to 5$	5	5	2,4,5,6
3	$4 \to 3$	3	3	2,3,4,5,6
4	$3 \to 1$	1	1	1,2,3,4,5,6

If we want to merge the $2nd$ ($2 \to 5$) and $3rd$ ($4 \to 3$) transmissions, with a combined cost of 6.22, into a single transmission, the transmitting node in the replacement transmission can be 2, 4 or 6 (elements of the set $\mathbf{NA}^{(0:1)}$). Further, the replacement transmission must reach at least nodes 5 and 3 (elements of the set $\mathbf{NA}^{(2:3)}$). From (2), it can be seen that possible replacement transmissions are $2 \to 3$, $2 \to 4$, $4 \to 5$ and $6 \to 3$. However, since the cost of each of these possible replacement transmissions (1) is greater than 6.22, it is not possible to improve the starting solution (13) by merging the $2nd$ and $3rd$ transmissions.

However, if we want to merge the $1st$ ($6 \to 4$) and $2nd$ ($2 \to 5$) transmissions, with a combined cost of 7.67, into a single transmission, the transmitting node in the replacement transmission must be 6 (the only element in the set $\mathbf{NA}^{(0)}$). Also, the replacement transmission must reach at least nodes 2, 4 and 5 (elements of the set $\mathbf{NA}^{(1:2)}$). The possible replacement transmissions in this case are $6 \to 1$, $6 \to 3$ and $6 \to 5$. Of these, only the cost of the transmission $6 \to 5$ (6.85) is smaller than 7.67. An improved solution can therefore be obtained by replacing the first two transmissions with $6 \to 5$.

$$\text{Improved solution} = \{6 \to 5, 4 \to 3, 3 \to 1\} \quad (15)$$
$$\text{Cost} = 14.93 \quad (16)$$

We now provide a high-level description of the e-merge procedure.

III. THE e-merge ALGORITHM

Starting with a processed starting solution[2], the e-merge algorithm applies procedure EM sequentially to check for an improved solution. If an improved solution is found, procedures MTR and ET are applied to eliminate multiple transmissions from any node and redundant transmissions. The procedure is then repeated on the improved tree, till no further improvement is possible. In the following description, we assume that $MTR(T)$ is a function which takes a tree T, applies procedure MTR on the tree and returns the updated tree. Similarly, $ET(T)$ is a function which takes a tree T, applies procedure ET on the tree and returns the updated tree.

```
/* Get a starting solution */
Let T be the starting solution.
K = no. of transmissions in T;

/* Process the starting solution before applying the
e-merge procedure. */
T ← MTR(T);
T ← ET(T);

/* Compute the number of transmission sets in the
tree which need to be checked for possible merging into a
single transmission. */
N_E = K − e + 1;

/* Apply procedure EM sequentially. */
if(K == 1)
  Stop. Print T;   /* End of procedure. */
else
  /* Find the cost of T and assign it to best_cost. */
  best_cost = cost of T;
  for(k = 1:N_E)
    take transmissions (k : k + e − 1) in T and check
    whether they can be replaced by a single transmis-
    -sion covering appropriate nodes;
      if(no replacement transmission found)
        /* Procedure terminates if all possible trans-
        -mission sets in T have been checked. Other-
        -wise, check the next transmission block. */
        if(k == N_E)
          Stop. Print T;   /* End of procedure. */
        end
      else
        /* Assign T to a temporary tree. */
        T_temp = T;
        replace transmissions (k : k + e − 1) in T_temp
        by the replacement transmission;
        T_temp ← MTR(T_temp);
        T_temp ← ET(T_temp);
        /* Find the overall cost of T_temp and assign it
        to new_cost. */
        new_cost = cost of T_temp;
        /* Replace T with T_temp only if there is reduc-
        -tion in cost. */
        if(new_cost < best_cost)
          best_cost = new_cost;
          T ← T_temp;
          /* Recompute K and N_E. */
          K = no. of transmissions in T;
```

[2]The starting solution, represented by an ordered sequence of transmissions, must correspond to a connected graph and should cover all intended destination nodes. As mentioned before, connectivity is ensured if the kth ($\forall k \geq 2$; for $k = 1$, the transmitting node is the source) transmitting node in the tree has been reached by any of the prior transmissions.

```
            N_E = K-e+1;
            if(K == 1)
                Stop. Print T; /* End of procedure. */
            else
                repeat for loop on the new T;
            endif
        endif
    endif
  endfor
endif
```

The e-merge procedure requires checking of at most $(K-e+1) + (K-e) + \cdots + 1 = (K-e+1)(K-e+2)/2$ transmission sets for possible replacement by a single transmission. This the case when an improvement is found in every pass (*i.e.*, in every execution of the **for** loop in the algorithm) of the algorithm, while examining the last block of 'e' successive transmissions in the tree, till the broadcast tree consists of a single transmission.

We now illustrate the procedure with an example.

IV. EXAMPLE, $e = 2$

We assume that node 1 is the source in Figure 1. Let the starting solution be the MST of the network.

$$\text{Starting solution } = \{1 \to 3, 3 \to 4, 4 \to 6, 6 \to 2, 2 \to 5\} \tag{17}$$

$$\text{Cost } = 21.50 \tag{18}$$

Nodes reached by each of the transmissions in (17) are shown in Table III.

TABLE III

Nodes reached by each transmission in (17).

k	$t \to d$	$\mathbf{NA}^{(k)}$	$\mathbf{NN}^{(k)}$	$\mathbf{NA}^{(0:k)}$
0	-	1	1	1
1	$1 \to 3$	3	3	1,3
2	$3 \to 4$	1,4	4	1,3,4
3	$4 \to 6$	3,6	6	1,3,4,6
4	$6 \to 2$	2	2	1,2,3,4,6
5	$2 \to 5$	5	5	1,2,3,4,5,6

Pre − processing of starting solution: The pre-processing step involves application of procedures MTR and ET on the starting solution. Since there are no multiple transmissions from any node and each transmission covers a node not covered by any other transmission (none of the elements in column 4 are empty in Table III), no improvement is possible by applying procedures MTR and ET on (17).

Iteration 1: We start with the first 2 transmissions in Table III. Possible replacement transmissions covering at least nodes 3 and 4 are $1 \to 4$, $1 \to 5$ and $1 \to 6$, the cheapest of which is $1 \to 4$. Table III is therefore temporarily updated as in Table IV. Since the $3rd$ transmission in Table IV is redundant ($\mathbf{NN}^{(3)} = \emptyset$), we can delete it, resulting in the following tree:

$$\text{Improved solution } = \{1 \to 4, 4 \to 6, 2 \to 5\} \tag{19}$$

$$\text{Cost } = 17.18 \tag{20}$$

Since the cost of the tree in (19) is smaller than the cost of the starting solution (18), we proceed to Iteration 2, with (19) as the starting solution.

TABLE IV

Temporary tree obtained after replacing the first two transmissions in (17) with the transmission $1 \to 4$.

k	$t \to d$	$\mathbf{NA}^{(k)}$	$\mathbf{NN}^{(k)}$	$\mathbf{NA}^{(0:k)}$
0	-	1	1	1
1	$1 \to 4$	2,3,4	2,3,4	1,2,3,4
2	$4 \to 6$	3,6	6	1,2,3,4,6
3	$6 \to 2$	2	$\emptyset$	1,2,3,4,6
4	$2 \to 5$	5	5	1,2,3,4,5,6

Iteration 2: Again, we start with the first 2 transmissions in (19). Possible replacement transmissions covering at least nodes 2, 3, 4 and 6 are $1 \to 5$ and $1 \to 6$, the latter being cheaper. The temporary tree with this replacement is shown in Table V. The cost of the temporary tree is 11.66, which is smaller than the cost of the tree in (19). We therefore break out of Iteration 2 and move to Iteration 3, the new tree being:

TABLE V

Temporary tree obtained after replacing the first two transmissions in (19) with the transmission $1 \to 6$.

k	$t \to d$	$\mathbf{NA}^{(k)}$	$\mathbf{NN}^{(k)}$	$\mathbf{NA}^{(0:k)}$
0	-	1	1	1
1	$1 \to 6$	2,3,4,6	2,3,4,6	1,2,3,4,6
4	$2 \to 5$	5	5	1,2,3,4,5,6

$$\text{Improved solution} = \{1 \to 6,\ 2 \to 5\} \tag{21}$$

$$\text{Cost} = 11.66 \tag{22}$$

Iteration 3: The only possible transmission which can replace the two transmissions in (21) is $1 \to 5$. However, since the cost of this transmission (14.92) is greater than the cost of the tree in (21), we accept (21) as the final solution.

Figures 2 and 3 show the starting solution and the final solution respectively.

V. SIMULATION RESULTS

We tested the e-merge procedure on 10, 25, 50, 75 and 100-node networks in a 5×5 grid. In each case, 50 networks were randomly generated and the tree powers averaged to obtain the mean tree power. 'α' was chosen to be equal to 2 for all cases. The mean tree powers for the BIP solutions are shown in column (2) in Table VI. The

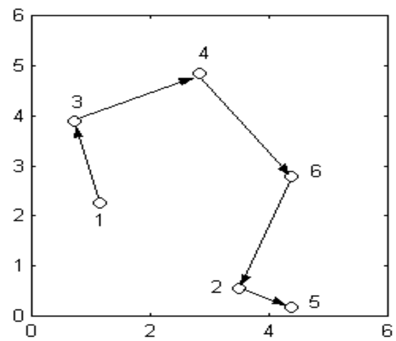

Fig. 2. Initial broadcast tree.

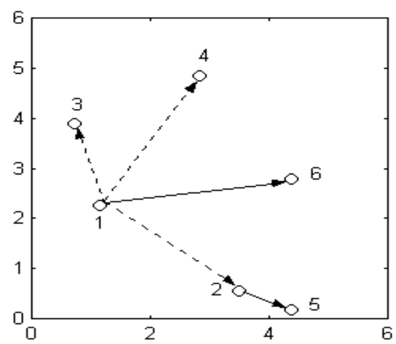

Fig. 3. Improved broadcast tree after applying the e-merge algorithm. Node 1 transmits to node 6, covering nodes 2, 3 and 4 in the process. Node 2 then transmits to node 5.

mean tree powers for the BIP solutions followed by the sweep algorithm discussed in [1] are shown in column (3). Column (4) lists the mean tree powers obtained by applying the e-merge procedure ($2 \leq e \leq 10$ for $N = 10$ and $2 \leq e \leq 25$ for $N = 25$, 50, 75, 100; N being the number of nodes in the network) on the BIP solutions. The figures in parentheses in columns (3) and (4) represent the percentage improvement in mean tree power over the BIP solutions.

It can be seen from Table VI that there is a significant improvement in tree power if the e-merge procedure is applied to the BIP solution as opposed to the sweep algorithm. The improvement is generally higher for networks with a large number of nodes (*e.g.*, $N = 75$ or 100, as opposed to $N = 10$). For each N, the number of times (out of 50 network instances) the e-merge algorithm was able to find a better solution than the sweep algorithm are given below:

- $N = 10$: 8 instances, out of 50.
- $N = 25$: 21 instances, out of 50.
- $N = 50$: 22 instances, out of 50.
- $N = 75$: 29 instances, out of 50.
- $N = 100$: 32 instances, out of 50.

It is evident that not only does the proposed procedure find better solutions than the sweep algorithm, it does so more often as the number of nodes in the network increases.

TABLE VI

Mean tree powers for BIP (column 2), BIP followed by sweep as in [1] (column 3), and BIP followed by e-merge (column 4). The figures in parentheses in columns (3) and (4) represent the percentage improvement in mean tree power over the BIP solutions.

N	BIP	BIP(sweep)	BIP(e-merge)
10	11.57	11.08 ($-$**4.23**%)	10.87 ($-$**6.05**%)
25	12.46	12.14 ($-$**2.57**%)	11.78 ($-$**5.46**%)
50	11.67	11.45 ($-$**1.89**%)	11.19 ($-$**4.11**%)
75	11.63	11.37 ($-$**2.23**%)	10.98 ($-$**5.59**%)
100	11.60	11.35 ($-$**2.16**%)	10.89 ($-$**6.12**%)

VI. CONCLUSION

In this paper, we have presented the e-merge algorithm, a heuristic tree-improvement procedure for improving minimum power broadcast trees in wireless networks. The procedure works by examining all possible blocks of 'e' ($e \geq 2$) successive transmissions in a broadcast tree and replacing them by cheaper single transmissions, if one exists. If an improvement is found, the procedure is repeated on the new tree, till no further improvement is possible. Simulation results show that better solutions are usually obtained if the proposed procedure is applied to the trees generated using the Broadcast Incremental Power (BIP) algorithm, in place of the sweep algorithm discussed in [1].

VII. ACKNOWLEDGMENT

This work is supported by the Advanced Information Systems Technology (AIST) program at the NASA Office of Earth Sciences.

REFERENCES

1) J.E.Wieselthier, G.D. Nguyen and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks", IEEE INFOCOM 2000, pp. 585- 594.
2) D. Bertsekas and R. Gallager, Data Networks, Englewood Cliffs: Prentice Hall, 1992.
3) Robert J. Marks II, Arindam K. Das, Mohamed El-Sharkawi, Payman Arabshahi & Andrew Gray, "Minimum Power Broadcast Trees for Wireless Networks: Optimizing Using the Viability Lemma", Proceedings of the IEEE International Symposium on Circuits and Systems, Scottsdale, Arizona, 2002.
4) Ivan Stojmenovic, Mahtab Seddigh and Jovisa Zunic, "Internal Nodes Based Broadcasting in Wireless Networks", Proceedings of the 34th Hawaii International Conference on System Sciences, 2001.